Aplikasi Pendeteksi Serangan pada HTTP Menggunakan N-Gram

Reinhard Ruben Rumare, Henning Titi Ciptaningtyas, Bagus Jati Santoso
Departemen Teknik Informatika, Fakultas Teknologi Informasi Institut Teknologi Sepuluh Nopember (ITS)
e-mail: henning@if.its.ac.id

***Abstrak*—Serangan digital saat ini jenisnya sangat banyak. Tiap hari jumlahnya juga selalu meningkat. Untuk mendeteksinya, ada banyak aplikasi yang menggunakan berbagai jenis metode untuk mendeteksi serangan – serangan yang ada. Riset – riset sebelumnya sudah menunjukkan bahwa analisis berlevel bite dari lalu lintas jaringan dapat digunakan untuk mendeteksi serangan dan analisis lalu lintas jaringan. Pada paper ini, penulis menggunakan 5 jenis teknik n-gram untuk mendeteksi serangan HTTP. Tujuannya adalah untuk membuat sebuah pertahanan pertama untuk serangan HTTP. Teknik – teknik n-gram ini dianalisa berdasarkan tingkat akurasi dan performanya. Hasil dari tes – tes yang dilakukan membuktikan bahwa teknik yang digunakan memiliki tingkat akurasi dan performa yang memuaskan.**

***Kata Kunci*— Analisis Byte , Analisis N-gram, Chi-squared Distance, Pearson Chi-squared Test, Serangan HTTP.**

I. PENDAHULUAN

KEMAJUAN teknologi dewasa ini semakin pesat. Teknologi yang baru berkembang semakin meningkatkan produktivitas manusia. Banyak kantor – kantor yang sekarang memperbolehkan karyawannya untuk langsung bekerja dari rumah, dan hanya perlu mengisi daftar hadir pada web kantor tersebut. Banyak pula toko –toko yang membuat usahanya berbasis internet menggunakan aplikasi web saja.

Banyaknya penggunaan web ini tentu saja memudahkan manusia untuk melakukan kegiatannya. Kita dapat melakukan kegiatan kita tanpa harus beranjak dari kursi kita. Namun, dengan meningkatnya penggunaan web untuk melakukan kegiatan kita, hal ini juga membuat sebuah celah bagi penjahat – penjahat virtual untuk mencuri data pribadi kita atau merusak web yang digunakan orang lain. Hal ini dapat berakibat fatal bagi para pengguna web.

Menurut *Symantec internet security threat* [1], sekitar 4500 jenis serangan terhadap web baru yang ditemukan setiap harinya. Server aplikasi web menjadi target yang umum untuk serangan – serangan tersebut karena server tersebut berkmounikasi dengan berbagai sistem lainnya.

Kebanyakan HTTP *traffic* hanya berisi karakter ASCII. Kita tidak menerima kode – kode yang bisa dijalankan pada paket HTTP yang kita terima. *Executable code* pada HTTP biasanya adalah sebuah penanda dari kemungkinan serangan injeksi *malware.* Distribusi byte dari ASCII jauh lebih terbatas dibandingkan dengan *executable code*, sehingga analasis dari distribusi byte pada paket HTTP bisa berguna untuk mendeteksi beberapa jenis serangan.

Teknik – teknik yang menggunakan analisis n-gram sudah pernah dilakukan sebelumnya untuk mendeteksi serangan HTTP [2] [3]. Pada [3] sebuah model n-gram diterapkan ke bite – bite yang terdapat dalam paket HTTP. Teknik n-gram ini kemudian diterapkan pada permasalahan pengklasifikasian file dalam ,dimana distribusi byte digunakan untuk mengklasifikan file *executable*, teks, atau multimedia. Paper [4] mengusulkan teknik mendeteksi malware dengan menggunakan χ^2 statistik.

Pada paper ini, penulis mempertimbangkan analisis n-gram menyerupai [5], dan χ^2 tes sejalan dengan [4]. Penulis menerapkan teknik – teknik ini ke permasalahan pendeteksian serangan HTTP. Tujuan dari paper ini adalah membuat sebuah pertahanan pertama efisien yang akan menyaring kebanyakan HTTP *traffic*, sehingga paket – paket yang berbahaya dapat dianalisa lagi dengan metode – metode yang lebih berat. Disini, penulis menyediakan hasil eksperimen dari metode – metode tersebut.

II. ANALISIS DAN PERANCANGAN

A. *N-gram*

N-Gram adalah sebuah urutan dari sejumlah n barang dari sebuah rangkaian teks atau kata – kata. Rangkaian ini dapat berupa apa saja, misalnya huruf, kata – kata, atau kalimat sesuai dengan apa yang mau kita gunakan. Sebuah N-gram berukuran 1 biasa disebut unigram, berukuran 2 biasa disebut bigram, berukuran 3 biasa disebut trigram. Ukuran yang lebih besar biasanya disebut sebagai *four-gram*, *five-gram*, dan seterusnya. [6]

Disini, n-gram ini digunakan untuk membuat pola yang muncul pada bite – bite pada paket dan frekuensinya.

B. *χ^2 distance*

Chi-squared Distance adalah sebuah metode untuk mencari jarak antara 2 histogram x=[x_1,x_2,....,x_n] dan y=[t_1,y_2,....,y_n]. Kedua histogram tersebut juga harus dinormalisasi dahulu, berarti isi dari kedua histogram tersebut harus berjumlah 1[7].

Perhitungan jarak antar 2 histogram tersebut (D(x,y)) dihitung dengan menggunakan rumus yang biasa digunakan untuk mencari *chi-square*, dapat dilihat pada persamaan (1)

$$D(X,Y) = \sum_{i=0}^{n} \frac{(X_i - Y_i)^2}{Y_i} \quad (1)$$

Keterangan :
n = Jumlah data unik pada histogram

X_i = Nilai normalisasi dari nilai dari x_i
Y_i = Nilai normalisasi dari nilai dari y_i

C. *Pearson χ^2 test*

Pearson Chi-squared Test adalah sebuah tes statistik yang diterapkan pada suatu kumpulan data untuk menilai seberapa besar kemungkinan munculnya perbedaan yang bisa dilihat antara kumpulan data tersebut secara kebetulan. Tes ini cocok digunakan untuk suatu kumpulan data yang besar. Tes ini adalah tes yang paling banyak digunakan diantara *chi-squared Test(*Sebuah prosedur statistik yang hasilnya di nilai dengan membandingkannya dengan distribusi *chi-square)* yang lainnya [8]

Tes ini menguji hipotesis *null* yang menyatakan bahwa distribusi frekuensi dari suatu kejadian yang dilihat dari contoh, mempunyai hasil yang konsisten dengan suatu teori distribusi lain.

Pada paper ini, hipotesa yang akan digunakan dapat dilihat pada persamaan (2).

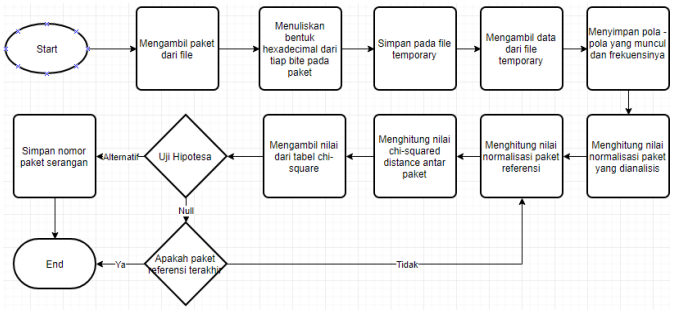
$$H_0 : D^2 \leq \chi^2(\alpha, b-1) \qquad (2)$$
$$H_1 : D^2 \succ \chi^2(\alpha, b-1)$$

Keterangan dari hipotesis ini adalah :

1. Hipotesa *Null* disini berarti paket adalah serangan, dan hipotesa alternatifnya adalah paket bukanlah serangan.
2. D^2 adalah *chi-square distance* antara 2 paket.
3. χ^2 adalah nilai dari chi-square table dengan nilai $\alpha = 0.05$ dan *degree of freedom* b-1.
4. a yang digunakan pada studi ini adalah $\alpha = 0.05$.
5. b adalah jumlah pola unik yang muncul pada paket referensi.

Analisis ini Dilakukan untuk mengetahui tingkat kemiripan dari paket yang akan dianalisa dan paket referensi. Jika paket yang dianalisa memliki kemiripan diatas batas, maka hipotesa alternatif berhasil, yang berarti paket tersebut bukanlah paket berbahaya. Jika paket yang dianalisa memiliki tingkat kemiripan dibawah dari batas, maka hipotesis nullnya berhasil, yang berarti paket tersebut adalah paket berbahaya.

D. *Deskripsi Umum Sistem*



Gambar 1. Desain Umum Sistem

Perancangan data adalah hal yang penting dalam aplikasi karena diperlukan data yang tepat agar aplikasi dapat berjalan dengan tingkat akurasi yang memuaskan. Aplikasi yang dibuat membutuhkan data referensi, data masukan dan data keluaran. Data referensi merupakan paket – paket bersih yang dipilih untuk menjadi pembanding apakah data masukan adalah paket berbahaya atau bukan. Data referensi ini berupa file berisikan byte dari paket – paket bersih. Data masukan berupa file hasil *sniffing* yang ingin dianalisa apakah paket – paket dalam file tersebut adalah serangan atau bukan. Data akhir adalah data lamanya aplikasi berjalan serta paket – paket mana saja yang dianggap sebagai paket serangan, setelah dilakukan pengecekan dengan data referensi.

E. *Desain Pembentukan n-gram*

Program akan mengambil pola – pola yang muncul pada bite – bite dalam paket dan kemudian akan menyimpan pola – pola tersebut dan frekuensi pola tersebut. Contoh N-gram pada bite dapat dilihat pada Tabel 1 dan Tabel 2.

Tabel 1.
Contoh Bite pada paket yang dianalisis

Pola	*Frekuensi*
00	0
09	11
11	5

Tabel 2.
Contoh Bite pada paket referensi

Pola	*Frekuensi*
00	3
09	5
11	7

F. *Desain Penghitungan Nilai Chi-square Distance*

Setelah didapatkan pola – pola yang muncul pada paket dan frekuensinya, program akan menghitung nilai normalisasi dari masing – masing pola yang terdapat pada paket yang ingin dianalisa, aplikasi kemudian akan menghitung nilai normalisasi dari masing – masing pola yang terdapat pada paket referensi. Kemudian, aplikasi akan menghitung *chi-squared distance* antara 2 paket tersebut.

Untuk paket yang terdapat pada Tabel 1 dan Tabel 2 untuk mendapatkan nilai chi-square distancenya, harus dihitung dulu nilai normalnya dahulu. Nilai normal dari tabel 1 adalah(00, 0), (09, 0.68), dan (11, 0.32). Sedangkan nilai normal dari table 2 adalah (00, 0.2) , (09, 0.33), dan (11, 0.47). Maka :

$$D^2 = \frac{(0-0.2)^2}{0.2} + \frac{(0.68-0.33)^2}{0.33} + \frac{(0.32-0.47)^2}{0.47}$$
$$= 0.2 + 0.237576 + 0.047872 = 0.485448$$

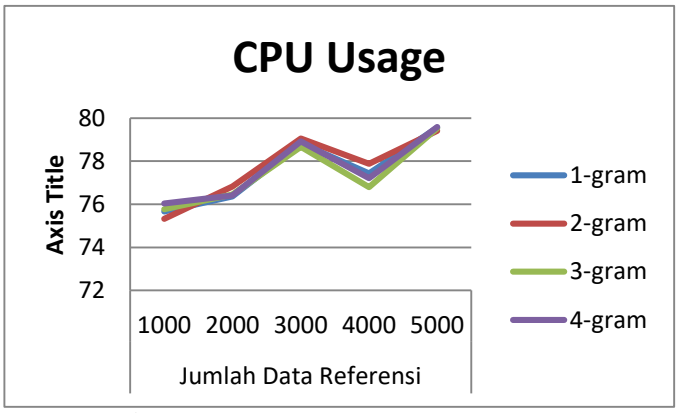
G. *Desain Pearson Chi-squared Test*

Program akan melakukan perbandingan antara nilai *chi-squared distance* antara paket yang dianalisa dan paket referensi dengan nilai dari table *chi-squared* dengan nilai $\alpha = 0.05$, dan *degree of freedom* b-1.

Dari perhitungan nilai chi-squared distane diatas, dapat dilihat bahwa nilai dari D^2-nya adalah 0.495449. Nilai dari $\chi^2(0.05,2)$ adalah 5.991. Karena $D^2 \leq \chi^2(0.05,2)$, maka paket yang dianalisa adalah paket berbahaya.

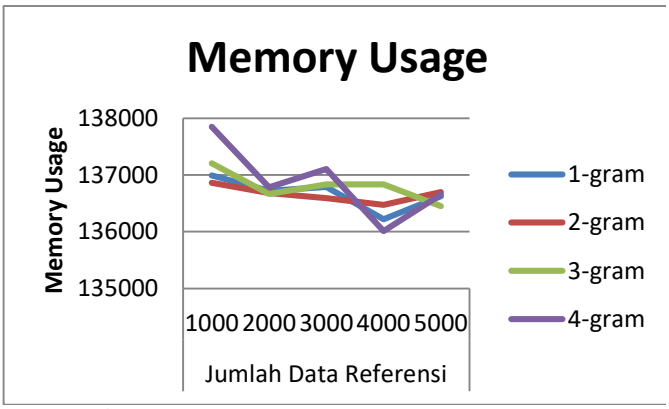
III. UJI COBA

Uji coba aplikasi dilakukan dengan melakukan analisa data – data yang sudah disediakan. Percobaan dilakukan dengan menggunakan 4 jenis n-gram dan 5 jumlah data referensi yang berbeda. Jumlah paket yang dianalisa pada uji coba ini adalah sebanyak 1000 paket. N-gram yang digunakan pada uji coba ini adalah 1-gram, 2-gram, 3-gram, dan 4-gram. Jumlah data referensi yang digunakan pada uji coba ini adalah 1000 paket, 2000 paket, 3000 paket, 4000 paket, 5000 paket. Pengujian yang dilakukan bertujuan untuk mengetahui tingkat efisiensi dan akurasi dari aplikasi ini.



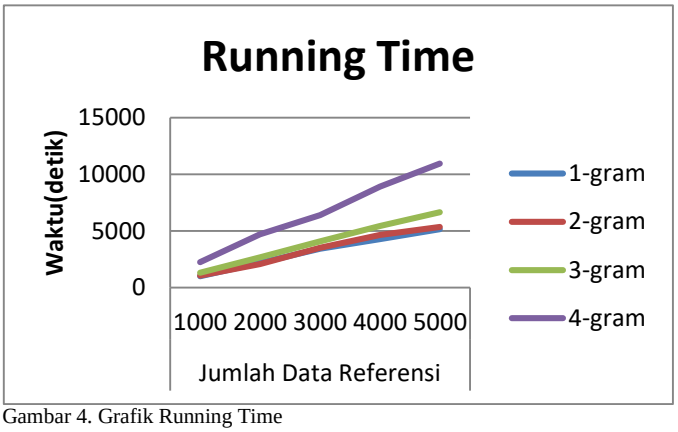
Gambar 2. Grafik Penggunaan CPU

Pada Gambar 2 dapat dilihat bahwa metode n-gram yang digunakan dan jumlah data referensi tidak berpengaruh besar terhadap penggunaan CPU.



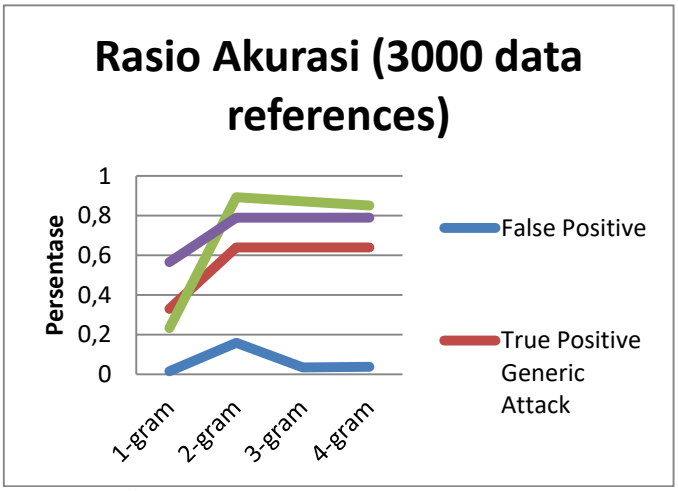
Gambar 3. Grafik Penggunaan Memori

Pada Gambar 3 dapat dilihat bahwa metode n-gram yang digunakan dan jumlah data referensi tidak berpengaruh besar terhadap penggunaan memori



Gambar 4. Grafik Running Time

Pada Gambar 4 dapat dilihat bahwa semakin banyaknya jumlah data referensi yang digunakan, semakin lama pula running time yang dibutuhkan oleh aplikasi untuk menganalisa paket. Meningkatnya jumlah n pada n-gram juga meningkatkan jumlah running time aplikasi ini.



Gambar 5. Grafik Rasio Akurasi

Pada Gambar 5 dapat dilihat bahwa rasio akurasi false positive 1-gram adalah yang paling baik sedangkan rasio akurasi false positive 2-gram adalah yang paling rendah. Namun, untuk rasio true positive, 2-gram selalu memiliki tingkat rasio akurasi yang paling tinggi pada ketiga serangan yang diuji coba, sedangkan 1-gram selalu memiliki rasio akurasi yang paling rendah.

IV. KESIMPULAN/RINGKASAN

Dari hasil uji coba yang telah dilakukan dapat diambil beberapa kesimpulan sebagai berikut :

1. Pengujian run time untuk aplikasi sudah berjalan dengan cukup baik, namun dirasa masih kurang optimal. Untuk nilai n yang bernilai 1 dan 1000 data referensi ,aplikasi membutuhkan waktu 1000 detik untuk menganalisa 1000 paket. Namun pada saat aplikasi menggunakan n = 4, aplikasi membutuhkan 2200 detik. Angka ini lebih dari 2x lipat waktu 1-gram. Sedangkan untuk data referensi 5000, 1 gram membutuhkan waktu 5000 detik, sedangkan 4 – gram

membutuhkan waktu 11000 detik. Dari sini dapat dilihat bahwa menggunakan 4-gram bukanlah cara yang optimal dalam penggunaan aplikasi ini.
2. Pengujian Penggunaan CPU dan penggunaan memori untuk aplikasi sudah berjalan dengan cukup baik. Dapat dilihat dari hasil pengujian bahwa jumlah n pada n-gram dan jumlah data referensi tidak berpengaruh banyak terhadap penggunaan CPU dan penggunaan memori
3. Pengujian False positive pada aplikasi berjalan dengan baik. Dari hasilnya dapat dilihat bahwa 1-gram memiliki tingkat rasio *false positive* terendah yang diikuti oleh 3-gram, 4-gram, dan terakhir oleh 2-gram. Disini dapat dilihat bahwa 2-gram memiliki tingkat false positive yang sangat tinggi senilai 15,9% dari 1000 data yang dianalisa. 3-gram dan 4-gram juga tidak terlalu jauh berbeda dengan hasil dari 1-gram.
4. Pengujian *True Positive* pada aplikasi berjalan dengan baik. Disini, kebalikan dengan hasil uji *false positive*, 2-gram memiliki hasil terbaik untuk semua jenis serangan yang diuji. Di pengujian ini dapat dilihat bahwa 1-gram memiliki nilai *true positive* yang sangat rendah dibandingkan dengan n-gram lainnya, terutama pada pendeteksian shellcode attack yang hanya memiliki tingkat akurasi 23,4%. 3-gram dan 4-gram juga tidak beberda terlalu jauh dengan hasil dari 2-gram.
5. Metode 3 – gram adalah metode terbaik untuk digunakan karena tingkat *akurasi false positive* yang rendah senilai 0.034, dan tingkat rasio *true positive* yang tinggi senilai 0.767838 untuk 3 jenis serangan yang diuji , dan juga *running time* yang tidak terlalu tinggi sebanyak 4 detik tiap paket dengan menggunakan 3000 paket referensi.

Adapun saran yang dapat dipertimbangkan untuk pengembangan atau penelitian lebih lanjut adalah sebagai berikut :

1. Mencoba menggunakan perhitungan jarak yang lain seperti ad-hoc chi square distance atau metode *pattern counting.*
2. Menggunakan data referensi yang lebih baik lagi.
3. Menyimpan hasil perhitungan nilai normalisasi data referensi sebelumnya agar pada saat penentuan serangan tinggal dipanggil saja nilainya, sehingga mengurangi waktu *run time* dalam menganalisa paket.

UCAPAN TERIMA KASIH

Tuliskan ucapan terima kasih dengan bahasa baku, misalnya, “Penulis A.F. (inisial nama mahasiswa) mengucapkan terima kasih kepada Direktorat Pendidikan Tinggi, Departemen Pendidikan dan Kebudayaan Republik Indonesia yang telah memberikan dukungan finansial melalui Beasiswa Bidik Misi tahun 2010-2014”. Penulis juga diperkenankan menyampaikan ucapan terima kasih kepada sponsor penyedia dana penelitian.

DAFTAR PUSTAKA

[1] Symantec.com, “Symantec Internet Security Threat Report,” *symantec.com.* [Online]. Available: http://www.symantec.com/threatreport/.

[2] M. S. M. and G. Z. A. Z. Broder, S. C. Glassman, “Syntactic clustering of the web,” vol. 29, pp. 8–13, 1997.

[3] G. W. S. and W. G. Cochran, *Statistical Method, Eigth Edition*. Iowa State University Press, 1989.

[4] “http://www.ling.upenn.edu/~clight/chisquared.htm,” 2015. [Online]. Available: http://www.ling.upenn.edu/~clight/chisquared.htm.

[5] P. R. and e. A, “McPAD: a multiple classifier system for accurate payload-based anomaly detection.,” *Comput. Networks Int. J. Comput. Telecommun. Netw.*, 2009.

[6] W. K. and S. S., “Anomalous payload-based network intrusion detection,” in *In: Proceedings of the 7th international conference on recent advances in intrusion detection*, 2004.

[7] T. A. and S. M, “Chi-squared distance and metamorphic virus detection,” *J. Comput. Virol. Hacking Tech.*, 2013.

[8] A. I. and L. K, “Classification of packet contents for malware detection,” *J. Comput. Virol.*, 2011.